A Fair Scheduling Scheme for a Time-Sensitive Traffic over the Dual-Channel Wireless Network

Junghoon Lee, Mikiyung Kang, Abhijit Saha
Dept. of Computer Science and Statistics
Cheju National University
Jeju, Republic of Korea
{jhlee,mkkang,abhijit298}@cheju.ac.kr

ABSTRACT
This paper proposes a message scheduling scheme capable of improving both throughput and fairness on the dual-channel wireless network. Based on the time-slotted access mechanism for each channel within a single cell, the coordinator schedules the transmission of the first channel by EDF (Earliest Deadline First) discipline that can maximize the actual throughput, while GDF (Greatest Degradation First) discipline is exploited for the second channel to improve the fairness factor. During runtime, with the support of channel estimation mechanism, the coordinator can dynamically switch the channel between the two scheduled streams. The simulation result shows that the proposed scheme improves the actual throughput by 7.2% compared with EDF, while the fairness is improved by 2.4% compared with GDF.

Categories and Subject Descriptors
C.2.1 [Computer-Communication Networks]: Network Architecture and Design; C.2.5 [Computer-Communication Networks]: Local and Wide-Area Networks; C.2.3 [Computer-Communication Networks]: Network Operations

General Terms
Dual-Channel

Keywords
Dual-channel wireless network, real-time communication, dynamic channel switch, fairness, deadline meet ratio

1. INTRODUCTION
With the increasing bandwidth in the wireless systems and rapidly growing demand for multimedia communications, the scheduling problem for real-time traffic has been studied intensively[1]. The real-time traffic class is modeled as a stream of packets, with each packet having a deadline, and the packet is of no use to the consumer application beyond the deadline[2]. The objective of scheduler is to transmit each packet before its deadline, and if not possible, to minimize the number of lost packets due to the deadline miss. In the wireline network, the example of scheduling policy includes EDD (Earliest Due Date) scheme and GPS (Generalized Process Sharing) model. These schemes are known to be not only optimal but also able to provide QoS (Quality of Service) guarantee in wireline networks.

However, the wireless link is not perfect and subject to errors, resulting in the burst of errors during which packets cannot be successfully transmitted over the link[3]. In addition, the channel errors are location-dependent, namely, a specific frequency may be accessible to some nodes but not to others, because the nodes are typically mobile stations with possibly different fading characteristics. Due to such reasons, scheduling policies known to be optimal in the wireline network are not necessarily the best policies in the wireless network.

Basically, at any time slot the scheduler selects the message to transmit through the shared wireless medium. For example, FEDD (Feasible Earliest Due Date) chooses the packet which has the earliest time to expiry only from the set of queues whose channels are marked good[1]. Even if the EDD can maximize network throughput, one problem with such greedy scheduling is the unfairness in resource sharing between nodes in the network. This is due to the fact that the node with the best channel conditions will always receive the biggest share of network resources, while the node suffering from bad channel conditions will not be able to be served.

On the other hand, to achieve fairness, a compensation function is indispensable to give the precedence to a lagged flow over the leading one. Typically, a flow is said to be lagging if, at any time instant, its queue length is shorter than that of its error-free counterpart at the same instant. Similarly, a flow is said to be leading if its queue length is longer than that of its error-free counterpart[4]. For example, GDF (Greatest Degradation First) discipline selects the most lagged flow when it returns to good[5]. However, these schemes provide no delay guarantees and pursuing only the fairness is meaningless for delay-sensitive applications. Accordingly, most of existing works attempt to improve fairness, trying to minimize the throughput loss[6]. The fairness issue is especially important with the recently emerging multimedia services in next-generation wireless networks.

In the mean time, the wireless network has an advantage that it can be easily duplicated, that is, a cell is able to
operate dual channels. For example, according to the IEEE 802.11b WLAN (Wireless Local Area Network) standard, a cell can have up to 3 channels in the 2.4 GHz band under 80 MHz reusable spectrum[7]. Multiple channels are also available in other wireless communication infrastructures capable of operating on multiple frequency bands. In this system, each node can be equipped with two or more (transmitter, receiver) pairs, giving the flexibility to transmit and receive simultaneously on both channels.

In this paper, we will focus our attention to the dual-channel system, as it can be seamlessly extended to multiple channel cases. The dual-channel system enables various ways to improve the specific system goal such as throughput, reliability, and fairness. In addition, the scheduler can allocate each channel with different policies, for example, one by EDD and the other by GDP and then dynamically change the allocation between the two channels according to the channel condition[8]. A node cannot send its message only when two channels are unreachable, resulting in the high possibility of meeting multiple scheduling goals simultaneously. With this assertion, this paper addresses the scheduling problem of achieving fairness among real-time flows with deadline constraints as well as maximizing the throughput of all the real-time flows over a wireless channel, based on the dual-channel network architecture.

The rest of this paper is organized as follows: Section 2 introduces backgrounds and related works on real-time message scheduling for dual-channel networks, and then Section 3 explains basic assumptions on network, message, and error models, respectively. Section 4 describes the proposed scheduling scheme in detail. After exhibiting the performance measurement results in Section 5, Section 6 concludes this paper and briefly describes future works.

2. RELATED WORKS AND BACKGROUNDS

As an approach to enhance the network throughput by applying the wireline EDD discipline to the wireless network, FEDD chooses the packet which has the earliest time to expiry only from the set of queues whose channels marked good[1]. The authors addressed that just looking at the deadlines is sufficient in most cases. The usefulness of this result lies in the fact that this allows us to implement simple scheduling policies without solving a very complex dynamic program which takes into account the channel and arrival process parameters at each time to schedule a packet.

Lee et. al have proposed an EDF (Earliest Deadline First) based scheduling scheme for the dual-channel WLANs[8]. EDF is another name of EDD. The basic idea is to split the given stream set into two identical ones and generate two distinct feasible schedules. Then rearrangement procedure makes two schedules as different as possible. At the runtime, the coordinator dynamically switches the scheduled polls between the two networks in response to the channel condition change. A node cannot send its message only when two channels are simultaneously unreachable. While this scheme can efficiently cope with channel errors in real-time communication, it didn’t consider the fairness issue at all.

While the above-mentioned approaches are based on the Gilbert error model, CDEDDB (Channel Dependent Earliest Due Date) policy is developed on top of a channel model having multiple different transmission rates[9]. It chooses to schedule the queue whose HoL (Head of Line) packet has the earliest deadline to expire and the best channel conditions, and in other words, the highest transmission rate, among all queues. This policy attempts to guarantee the targeted delay bounds in addition to exploiting multiuser diversity to make best utilization of the variable channel capacity.

In addition, M. Ada et. al have proposed LFF (Lagging Flow First) scheme that tries to favor lagging flows in a sophisticated manner[5]. The basic idea of this algorithm is to reserve time slots for packets to the latest possible time subject to meeting their deadlines in a decreasing order of flow degradation values. This algorithm makes reservation from the most lagging flow to the least lagging flow, pursuing both fairness and throughput simultaneously. However, sometimes one goal may be sacrificed to achieve the other goal in a single network, but this situation can be overcome in the dual-channel network.

3. SYSTEM MODEL

3.1 Network Model

Figure 1 shows a wireless-cum-wired network scenario in which a fixed node is connected with a BS (Base Station) through a wired link. The link is overprovisioned so that no packets are dropped at its end. We consider a time-slotted cellular system in which the service area is divided into multiple cells, and each is served via a BS. The wireless communication infrastructure may be WLAN, HDR (High Data Rate), CDMA (Code Division Multiple Access), and so on[8, 10]. In a cell, a centralized scheduler at the base station controls the downlink scheduling, whereas uplink scheduling uses an additional mechanism such as polling to collect transmission request from mobile terminals. In the hot spot multimedia model, most flows are downlink, and the flow arrives periodically from the remote server outside the cell.

To provide the time-slotted access on a channel, the time axis is divided into fixed size slots during which data transmission of a scheduled station is carried out using all resources available[11]. The slot length, say L, is as large as the basic unit of wireless data transmission and every traffic is also segmented to fit the slot size. In addition, the target infrastructure consists of dual channels, and slots are synchronized across the channels. That is, two channels simultaneously start their slots in phase[8].

Nowadays, the wireless channel is capable of providing link adaption scheme based on the multiple transmission rates[12, 13]. In this case, we believe that EST (Earliest Slack Time) discipline can be exploited instead of EDD, because the estimated transmission time keeps changing according to the current transmission rate. However, the scheduling in this environment is another problem and the multiple
we can calculate per-slot overhead and merge it into packet loss rate is above its acceptable loss rate. That is, a flow is defined to be lagging if its period coincides with its period at time 0. Otherwise, the packet is considered to be lost.

3.2 Error Model

Each station is associated with a channel link which has either of two states, namely, error state and error-free state at any time instant. A channel link is defined between each mobile and the BS, and it can be modeled by the two-state discrete-time Markov chain also known as the Gilbert-Elliot channel[3]. We can denote the transition probability from state good to state bad by p and the probability from state bad to state good by q, as shown in Figure 2. The pair of p and q representing a range of channel link conditions, has been obtained by using the trace-based channel link estimation. The average error probability and the average length of a burst of errors are derived as \( \frac{p}{1-q} \) and \( \frac{1}{1-q} \), respectively.

A packet is received correctly if the channel link remains in state good for the whole duration of packet transmission. Otherwise, it is received in error. Channel links between the BS and respective stations are independent of one another in their error characteristics. Correspondingly, while error-free transmission may be possible between a given node and the BS, transmission between another node and the BS may be corrupted by errors. For each packet, we must get feedback information on the transmission status of this packet before sending the next. The protocol indicates transmission success by sending back an ACK to the transmitter and transmission failure is detected by the lack of an ACK[7].

3.3 Message Model

The traffic of real-time data is typically isochronous (or synchronous), consisting of message streams that are generated by their sources on a continuing basis and delivered to their respective destinations also on a continuing basis[14]. This paper follows the general real-time message model which has n streams, namely, \( S_1, S_2, \ldots, S_n \). We refer to a message stream as a flow and a message instance as a packet. Examples of such packets include constant bit-rate digitized voice and video data packets. After all, each \( S_i \) generates a message not more than \( C_i \) at each beginning of its period \( P_i \), while the first message of each stream arrives at time 0.

Each packet must be delivered to its destination within \( D_i \) time unit from its generation or arrival at the source, otherwise, the packet is considered to be lost. \( D_i \) usually coincides with \( P_i \) to ensure that the transmission completes before the generation of the next message. Additionally, a packet loss rate up to \( e_i \) is acceptable, and this parameter is the important criteria to determine whether a flow is lagging or leading. That is, a flow \( f_i \) is defined to be lagging if its packet loss rate is above its acceptable loss rate \( e_i \). Each stream belongs to a specific station, so if a slot is assigned to a stream, it means that BS should poll that station at the slot time.

As for the protocol overhead \( \Delta \), it is fixed for a stream set, we can calculate per-slot overhead and merge it into \( C_i \). If we let \( \Delta_p \) be the per-slot overhead, the modified transmission time, \( C_i^p \), can be calculated iteratively with the following equation until \( C_i^{p+1} \) becomes equal to \( C_i \).

\[
C_i^{p+1} = C_i^p + \left[ \frac{C_i^p}{L} \right] \Delta_p
\]

As a result, if we think \( \Delta \) is absorbed into \( C_i \), \( \Delta \) can be assumed to be 0, enabling us to concentrate on the problem of slot allocation.

4. SCHEDULING ALGORITHM

To begin with, we consider wireless systems with mechanisms to make predicted channel conditions known to the base station as is commonly the case with technologies such as HDR, GPRS (General Packet Radio Service), and so on. The particular mechanism employed by a system depends on the corresponding communication standard. The faster and more precise the channel quality can be predicted, the better the scheduler can incorporate this information into its decision as to which packet to schedule next. In short, we assume that base station has the current channel state information of each node.

The scheduler maintains two guaranteed queues and one best-effort queue. Each guaranteed queue is associated to each wireless channel while each entry of this queue is mapped to the time slot of the channel. The best-effort queue maintains the packet that cannot enter the guarantee queue because it has lower priority according to the given criteria. Namely, this queue contains the overflowed packets. In addition, if a packet has failed in transmission but its deadline is not expired, it is moved to best-effort queue to wait another chance of retransmission if possible.

Scheduling algorithm consists of two parts, namely, packet schedule and runtime operation. When a new packet is generated or becomes available to be scheduled, packet scheduler reserves a time slot for the packet by reordering the waiting queue according to the deadline. In this procedure, some packets are reordered inside the guaranteed queue or moved to the best-effort queue. Then the runtime scheduler makes a decision at every slot time boundary based on the current channel status of two networks.

For the first guaranteed queue, at each scheduling decision time, the packet with the earliest deadline is scheduled first and ties are broken randomly. In other words, packets are scheduled according to the EDF policy. For the second guaranteed queue, a packet belonging to the flow with the greatest degradation value is scheduled first. Ties are also broken randomly. Intuitively, this algorithm is expected to maximize the fairness. These schemes are designed to pursue their own goals, but the dual-channel network enables to achieve more than expected with the inherent fault-tolerant nature.

During runtime, the scheduler adapts the guaranteed schedule based on the respective channel status of each node. Let’s assume that BS is to start slot \( i \) which is originally allocated to \( A \) on channel 1 as well as \( B \) on channel 2, namely, \( A, B > [8] \). BS first checks the channel condition from itself to \( A \) and \( B \) on all two channels. Thus each slot should inevitably contain two probing latencies. Table 1 shows the probing result and corresponding actions. As shown in row 1, BS can reach \( A \) on channel 1 and also \( B \) on channel 2. BS polls each station as scheduled. In row 2, all connections from BS are good except the one to \( B \) through channel 2. If
we switch $A, B >$ to $< B, A >$, both streams can successfully send their messages. Otherwise, only $A$ can send on channel 1, so in this case, we can save one transmission loss. Row 8 describes the situation that BS can reversely reach only on channel 2 while $B$ on channel 1. By switching polls between the two channels, BS can save the 2 transmissions that might fail on ordinary schedule.

The polling table in Table 1 has some entries marked as ‘-‘, which means BS cannot poll $A$ or $B$. In this case, it seems better to poll another station, and the packet in the best-effort queue is selected. In this case, neither $A$ nor $B$ can be selected.

<table>
<thead>
<tr>
<th>No.</th>
<th>Ch1−A</th>
<th>Ch2−B</th>
<th>Ch1−B</th>
<th>Ch2−A</th>
<th>Ch1</th>
<th>Ch2</th>
<th>save</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Good</td>
<td>Good</td>
<td>X</td>
<td>X</td>
<td>A</td>
<td>B</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Good</td>
<td>Bad</td>
<td>Good</td>
<td>Good</td>
<td>B</td>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Good</td>
<td>Bad</td>
<td>Good</td>
<td>Bad</td>
<td>A</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Good</td>
<td>Bad</td>
<td>Bad</td>
<td>X</td>
<td>A</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>Bad</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>B</td>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>Bad</td>
<td>Good</td>
<td>Good</td>
<td>Bad</td>
<td>-</td>
<td>B</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>Bad</td>
<td>Good</td>
<td>Bad</td>
<td>X</td>
<td>-</td>
<td>B</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>Bad</td>
<td>Bad</td>
<td>Good</td>
<td>Good</td>
<td>B</td>
<td>A</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>Bad</td>
<td>Bad</td>
<td>Good</td>
<td>Bad</td>
<td>B</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>Bad</td>
<td>Bad</td>
<td>Bad</td>
<td>Good</td>
<td>-</td>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>Bad</td>
<td>Bad</td>
<td>Bad</td>
<td>Bad</td>
<td>-</td>
<td>-</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1: Channel status and transmission

| X: don’t care |

5. PERFORMANCE MEASUREMENT

This section measures the performance of the proposed scheme in terms of deadline meet ratio (DMR) and fairness factors via simulation using ns-2 event scheduler[15]. In the experiments, every time variable is aligned to the slot time. For example, a stream $(4,1)$ means that $P_i$ is $4L$ while $C_i$ is as large as $IL$. The DMR is the most important performance criteria in real-time communication and it is analogous to the actual network throughput. The DMR can be calculated as the fraction of timely delivered real-time packets to all generated packets. Namely,

$$DMR = \frac{\sum_{i=1}^{n} M_i^u}{\sum_{i=1}^{n} M_i}$$

, where $M_i$ is the number of packets that stream $S_i$ was supposed to deliver and $M_i^u$ is the number of packets that stream $S_i$ is actually successfully delivered.

To measure the fairness, we use $\epsilon_{max}$ which reflects the maximum difference between the most lagging stream and the most leading stream. Namely,

$$\epsilon_{max} = \max(|\epsilon_i - \epsilon_j|)$$

, where $\epsilon_i = 1 - \frac{M_i^u}{M_i} - \epsilon_i$. In the subsequent experiments, all $\epsilon_i$’s have the same value to simplify the measurement without the loss of generality.

Figure 3 shows DMR according to the standard deviation (STV) of packet error rate (PER). We selected the STV for the experiment parameter to differentiate the PER of each message stream. PER is the function of bit error rate and packet length. We set the number of stream sets to 5 and their load to 0.7. Additionally, the load, $U$, is calculated as $\sum_{i=1}^{n} C_i$. The average PER is 0.9 and each stream has different error rate with the parameterized STV. Figure 3 demonstrates that the DMR of the proposed scheme is higher than those of EDF and GDF due to the advantage of dynamic channel switching. As this experiment measures the system-wide DMR and the average PER remains constant, DMR does not change for EDF, GDF, and the proposed scheme. In addition, as the number of successful transmission is counted irrespective of which stream the transmission belongs to, EDF and GDF show almost same performance.

Figure 4 plots the fairness against the STV of packet error rate. In this figure, $\epsilon_{max}$ of the proposed scheme is smaller than those of GDF and EDF by 2.4% and 4.2%, respectively. GDF shows a little bit better fairness compared with EDF scheme, as it explicitly gives a precedence to the lagged stream and hence more chance of retransmission. Figure 3 and Figure 4 indicates that neither DMR nor fairness is sacrificed on dual-channel networks.

Figure 5 plots the DMR according to the offered load. In this experiment, the STV of error rate is set to 2.0, while the average packet error rate is 0.9. The proposed scheme and EDF would meet the deadline of every packet on the error-free channel, as long as the offered load is less than or equal to 1.0. However, GDF misses some deadlines when the offered load exceeds 0.7, because it cannot optimally schedule the real-time packets. The proposed scheme improves the DMR by 7.2% compared with EDF.

Finally, Figure 6 measures the effect of offered load to the fairness. The proposed scheme shows small and stable deviation in the $\epsilon_i$, as it can succeed in almost all transmissions. Though $\epsilon_{max}$ increases as the offered load increases, its amount is extremely insignificant. After all, the proposed scheme can reduce $\epsilon_{max}$ by around 2% compared with GDF schedule.
6. CONCLUDING REMARKS

This paper has proposed a message scheduling scheme capable of improving both throughput and fairness on the dual-channel wireless network. Within a cell, each channel is accessed via the time-slotted manner, and the coordinator schedules the transmission of each flow. For the first channel, the coordinator allocates the slot by the EDF discipline that can maximize the actual throughput. As contrast, for the second channel, GDF schedule is generated to improve the fairness factor. During runtime, with the support of channel estimation procedure, the coordinator can dynamically select the error-free channel for the two scheduled streams. Hence, the stream fails to transmit its message only when both channels are simultaneously inaccessible. The channel switch is able to improve the possibility of meeting the scheduling goal. Simulation results shows that the proposed scheme always outperforms EDF and GDF in throughput and fairness, as it does not sacrifice one goal for the other. Specifically, the channel switching improved the actual throughput by 7.2 % compared with EDF, while the fairness is improved by 2.4 % compared with GDF. Dynamic channel switching scheme can be extended to integrate some independent multiple goals.

As a future work, we are investigating a scheduling scheme for the time-sensitive message streams on wireless channels supporting multiple transmission rates. We believe that such a scheduling scheme will help a real-time contiguous queries to expand its usage range to the highly mobile telematics system.

7. ACKNOWLEDGMENTS

This research was supported by the MIC, Korea, under the ITRC support program supervised by the IITA (IITA-2006-C1090-0603-0040).

8. REFERENCES